**Report Project 1 Part 3**

**Student: Maria Alejandra Zapata Montano**

**A screenshot of a game

AI-generated content may be incorrect.**

**1)Problem formulation**

Terrain and resource tiles in a 5x5 grid. Players A and B take turns moving up, down, left, and right by one step. Every participant gets a fixed capacity backpack that holds two items. If a player steps onto a resource tile and has space, they pick it up; if they step onto their own base, they deliver everything they’re carrying. Game ends when all resources have been delivered. Whoever delivered more is the winner.

State:

* Positions of A and B
* Backpack contents for A and B
* Delivered counters for A and B, plus a global delivered total
* Remaining resources
* Whose turn it is

Actions:

* Legal actions are the 4 neighborhood moves that stay on the board.
* The transition does, in order:
  1. move to target cell
  2. collect if a resource is present and bag has room
  3. deliver if standing on the player’s base
  4. switch turn

Utility:

Zero sum from A’s perspective.

* Terminal states: all resources delivered. We use the exact utility.
* Non terminal states: we use a heuristic score.

**2)Heuristic design**

We wanted to design a zero-sum estimate of who’s ahead right now.

**Signals we used:**

* Possession (delivered + in bag): nearest to actual points; should dominate.
* closest needed resource (when empty): be closer to something you can actually pick up.
* Distance to base (when carrying).
* Opponent pressure (cheap): subtract the opponent’s versions of the two proximity terms.
* Tiny step penalty: To breaks ties and discourages stalling.

**Final heuristic:**

The formula **Eval(s)** estimates who is ahead. **del\_A**/**del\_B** are items already delivered by A/B (banked points). **bag\_A**/**bag\_B** are items currently carried, valuable because they’re close to becoming points. **POSSESSION\_W** is a large weight so this (delivered + carried) score dominates. **NearRes\_A**/**NearRes\_B** measure closeness to the **nearest remaining resource**, and **HomePull\_A**/**HomePull\_B** measure closeness to each player’s **own base when carrying**. **STEP\_PENALTY** is a small penalty per step to break ties and discourage stalling. Each component is taken as A minus B,making the score **zero sum**: positive favors A; negative favors B. We used heuristic with depth 5.

**3)Results**

* **Map A**

**A screenshot of a game

AI-generated content may be incorrect.A screenshot of a game

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.**

In the screenshots, the alpha beta/Minimax agent behaves as intended: against Random (top-right) it finishes the game cleanly with A 4–0 deliveries and 0 remaining tiles, with sub-millisecond last-move time, evidence that alpha–beta is giving the same choices as plain minimax while expanding fewer nodes and running faster. The Minimax vs Minimax match (bottom) is a sensible endgame: A has 3 delivered, B has 2 delivered and 1 in the bag, and no resources remain on the map, so it’s purely a race to base; the heuristic drive both agents to cash out rather than oscillate. Overall, results match the analysis: alpha beta improves efficiency without changing decisions.

* **Map B**

**A screenshot of a game

AI-generated content may be incorrect.A screenshot of a game

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.**

On Map B we see the same pattern as Map A but with tougher terrain: the top-left is a fresh start (A vs Minimax, 6 resources on board), while the top-right shows **GAME OVER vs Random** with **A wins** and **0 remaining,** again confirming the alpha–beta/Minimax agent converts pickups into deliveries quickly. The bottom snapshot is a Minimax vs Minimax: **A delivered 3 B delivered 3** and **no resources remain;** Overall, Map B results reinforce the analysis: alpha–beta yields the same choices as plain minimax but faster, dominates Random.

* **Map C**

**A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a game

AI-generated content may be incorrect.**

**A screenshot of a game

AI-generated content may be incorrect.**

On Map C, we see the same behavior: the start screen (top-left) shows a fresh board vs Random, and the finish (top-right) shows GAME OVER with A 4–2 B deliveries and 0 remaining, so the alpha beta agent still converts quickly even on harder terrain. In self-play (bottom), the game also ends cleanly with A 4–2 over a Minimax opponent. Overall, Map C reinforces the pattern: alpha–beta yields fast decisions, dominating Random and producing decisive results in Minimax-vs-Minimax.

**4)Discussion**

The heuristic used was so good because of how solid and simple it is:

* The **possession** term (delivered + carried) gives a strong direction.
* The **home pull** when carrying removes indecision.
* The **context aware proximity** avoids fighting between two goals.
* The **step penalty** push game to finish.

Also, across Maps A, B, and C, we met the goal of implement a computer search agent using minimax with alpha–beta pruning: it consistently **beats Random cleanly** (4–0 with zero tiles remaining), while **minimax vs minimax** produces stable endgames that hinge on who can bank first. Alpha–beta delivers the **same choices as plain minimax** but with far **fewer nodes and lower latency**.

Regarding the depth of heuristic: the deeper you search, the smarter the moves, but the tree blows up fast. With alpha beta and good move ordering, **d=5** was the best for us: noticeably better play, **d=3–4** is super fast and “good enough” for most boards. Pushing to **d≥6** helps in tricky endgames, but costs a lot more nodes. Thus, more depth is better quality, but it depends on your resources. For this game, our best option was d=5.